E-Model based prioritization of Multiple VoIP sessions over 802.11e

Padraig O Flaithearta, Hugh Melvin
Disc. Of Engineering and Informatics
NUI Galway
Galway, Ireland
p.oflaitheartal@nuigalway, hugh.melvin@nuigalway

Abstract — *The NS-2 Network Simulator is an open source tool that is widely used in networking research. Due to its open source, extendable nature, NS-2 provides support for simulation of many protocols over networks including 802.11e. The ITU-T E-Model is a standard quality measurement tool for assessing multimedia over networks. Our research involves building an NS-2 module extension that will dynamically prioritize certain VoIP sessions over others and attempt to equalize QoS (Quality of Service) for all sessions within an Access Category. Prioritization will be based on R-factor values from the E-Model. As part of our overall research, this paper details some of the improvements that can be brought about by prioritizing certain VoIP calls over others within a wireless MAC environment based on non-wireless MAC delays.*

1 INTRODUCTION

As the demand for real time business and personal applications increases over WiFi networks, the potential savings that VoIP in particular can provide are contributing to the demand for improvements in voice traffic (QoS). The 802.11e protocol was developed with time sensitive applications in mind, however, severe congestion leading to unacceptable delays and packet loss can still occur. The primary goal of our research is to investigate how synchronized time implemented in end terminals can aid in optimizing 802.11e parameters in order to improve QoS for VoIP. We are employing a combination of building an extension to the NS-2 network simulator and, simultaneously, building a practical test bed to back up our results.

This paper details simulations carried out on NS-2 that show the improvements that can be brought about for VoIP QoS by tuning 802.11e EDCA MAC parameters, based on synchronized time. In our NS-2 extension, which is still under development, the algorithm to dynamically vary parameters is informed by the E-Model R-factor, based on per-session each-way delay information. It will implement and maintain, where appropriate, an equalization of R-factors within the Voice access category (AC_VO). Although this paper presents simulation results for an all-VoIP scenario as the use case example, this extension is applicable to any EDCA access category or traffic makeup, and future work will include experiments using different traffic characteristics.

2 BACKGROUND

2.1 The ITU-T E-Model

Quality of Service can be defined as a metric that measures the overall user satisfaction with network performance. QoS is application dependent and the QoS requirements may vary widely depending on the application. There are different metrics available to quantify the QoS for VoIP applications. One widely used metric is the ITU recommendation G.107 E-Model.

The E-Model outputs a number rating called the R-factor, which is computed as a function of delay, packet loss, equipment impairment factors, and user quality call expectation. The value is calculated as follows:

$$\mathbf{R = Ro - Is - Id - Ie,eff + A} \qquad [6]$$

where **Ro** is the basic signal-to-noise ratio (received speech level relative to circuit and acoustic noise). **Is** accounts for the impairments which occur with the voice signal. **Id** is the sum of all impairments due to delay and echo effects. With respect to M2E delay, **Id** (in presence of good echo cancellation) can be approximated by 10 units per 100msec up to 150msec, and by 12-13 units per 100msec thereafter. The **Ie,eff** is the effective equipment impairment factor, taking into account the codec and its tolerance to packet losses [1]. **A** is a bonus factor that models the user expectation of the technology employed and can be used to take into account the fact that the user will tolerate some decrease in transmission quality when using certain technologies.

The major factors affecting QoS in a wireless network are throughput, packet loss, packet delays and jitter. Some of the main problems associated with achieving good QoS in a wireless environment are that the wireless channel can change with time and so it is possible that links between two nodes could break during a voice session.

The R-factor values are defined and categorized as shown in Table 1.

90 ≤ R < 100 Best: Very satisfied
80 ≤ R < 90 High: Satisfied
70 ≤ R < 80 Medium: Some users dissatisfied
60 ≤ R < 70 Low: Many users dissatisfied
50 ≤ R < 60 Poor: Nearly all users dissatisfied

Table 1: Definition of Categories of Speech transmission Quality

Our NS-2 module will employ an R-factor based system as a measurement of quality for dynamically determining the necessary parameters in order to carry out its functionality. When the network is monitored during a simulation, an R-factor will be calculated for each individual session, and these R-factor values will be then used to calculate which sessions require prioritization over other sessions.

2.2 The Network Simulator NS-2

This paper provides results from NS-2 simulations which are essentially set up using a scripting language to configure network topology and traffic, however, our extension which is under development requires development at a lower level. The NS-2 is an object oriented, discrete event simulator written in C++, with an OTcl (Object-Oriented Tool Command language) interpreter as a frontend [3]. A set of discrete events are executed in order by a simulator object. OTcl is its primary command and configuration language, it implements network protocols such as TCP and UDP over wired and wireless networks, and also traffic behavior such as FTP, Telnet, Web, CBR (constant bit rate) & VBR(variable bit rate). NS-2 implements router queue management mechanisms, multicasting and some of the MAC layer protocols (which include 802.11b MAC layer specification) for LAN simulations. The NS-2 package also includes tools for analysis and display of the simulation results including NAM(Network Animator).

The simulator performs two main functions, it performs detailed and efficient simulations of protocols, for which C++ is required for handling bytes, packet headers and efficient algorithm implementation. The other function is allowing users to vary parameters quickly to compute different outputs from simulations, which is controlled by the scripting language Tcl.

Given the two language structure of NS-2, certain tasks can be performed in a Tcl script, such as configuring network topology or assigning traffic characteristics, whereas more complex tasks involve editing OTcl and C++ code which requires recompilation.

The goal of our NS-2 module is to dynamically tune 802.11e EDCA parameters for wireless nodes (within an Access Category) that are hosting real time traffic sessions, in order to equalize the overall M2E delay for all sessions where appropriate. This will be done based on information provided by the E-Model.

3 VOIP OPTIMIZATION BASED ON TIME SYNCHRONIZATION

The overall goal of our research is to investigate what improvements can be brought about for VoIP when implementing time synchronization across wireless networks. This is essentially composed of two parts; developing an extension for NS-2, and by running real world experiments in order to validate results.

3.1 Adaptive algorithm

Our extension module involves adapting EDCA parameters according to an algorithm that computes the optimal settings based on each way delay calculations. This algorithm will calculate in real-time the R-factor (including delay) for each individual session, and then use this information to tune parameters in order to gradually equalize the overall R-factors for all sessions (Fig. 2/3).

The first goal of our algorithm will be to check whether a session has a large M2E delay relative to other sessions, and if so, it will aim to reduce the delay for that session(s), once it does not compromise the QoS of the remaining sessions. The objective thereafter will be to concurrently pursue and achieve a state of equalization of R-factors for all remaining sessions.

Depending on the scenario and the desired outcome, there are varying combinations of EDCA parameters that can be tuned. In a scenario where there are multiple VoIP sessions running on an 802.11 network (such as in Fig. 4), and the goal is to equalize M2E delays for all sessions, as mentioned earlier, the algorithm uses the R-factor to measure the QoS for each session. Where R-factor is below 70 for example, a user would experience a “medium” level of QoS, and some users would be dissatisfied according to the E-Model. QoS for individual sessions would then be calculated in order to determine how 802.11e EDCA parameters will be tuned. We are building this into our module as the basis of our delay equalization algorithm.

We are using the 802.11e extension in [5] to simulate the EDCA MAC. We are developing a C++ procedure that implements our algorithm to periodically adapt 802.11e parameters for each individual wireless node, based on R-factor information associated with the real-time session being hosted on that node.

At the beginning of a simulation the MAC requests the 802.11e parameters for each priority. The AIFS, CW_MIN, CW_MAX and TXOP Limit values are assigned to each queue before traffic sessions begin to transmit. After a certain period, R-factor values will be calculated, and 802.11e parameters will be re-assigned if required. This process will be repeated periodically for the duration of the simulation.

As outlined by ITU-T recommendation G.114[4], the acceptable limit of M2E delay for VoIP applications is approx 150ms. Although wireless MAC delays for multiple VoIP sessions through a single access point will be roughly equivalent, the overall M2E delay will typically vary greatly. For example, some calls take place over a LAN, with typical network delays < 10ms, and others over long distance with network delays > 100ms (Fig. 1).

Fig. 1: Different Baseline Delays

With such latter network delays the total M2E delay (Sender including MAC contention, Network and Receiver) of packets could be close to, or may exceed the ITU-T G114 recommendation of 150msec.

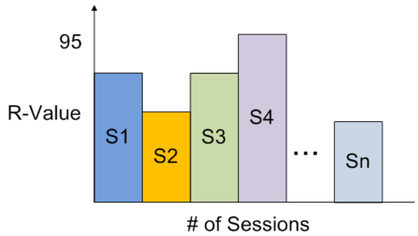


Fig. 2: One way delays can vary for VoIP sessions, which will in turn effect the R-Value

All sessions within the 802.11e EDCA voice category have similar delays at the wireless MAC layer. We propose that if the precise M2E delay information for each VoIP session is known in both directions, EDCA parameters can be configured differently in real-time between VoIP sessions in order to optimize the channel delays, so that the sessions with the higher M2E network delays receive higher priority treatment at wireless MAC level relative to other VoIP sessions with lower delays. Essentially priority is assigned to voice sessions within the voice access category.

For a session that has a small M2E delay, packets can afford to wait for a longer time at the wireless MAC layer, on the condition that they don't cause a significant degradation in QoS for that session and that they are delivered within 150ms whereas for a session that has a long network delay, packets have a shorter contention delay at the wireless MAC layer due to prioritization, thus reducing the overall M2E delay for that voice session.

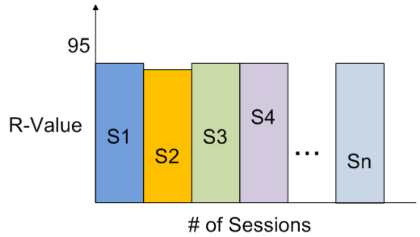


Fig.3: With Knowledge of each way delays, we can achieve acceptable QoS by equalizing the R-factor among Voice Sessions by dynamically tuning EDCA parameters

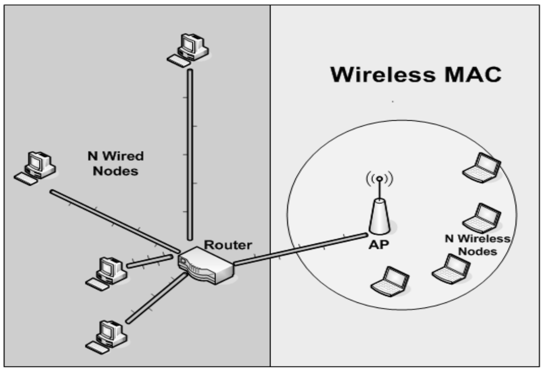


Fig. 4: Example scenario: Multiple VoIP sessions in a network with varying one way delays

Our methodology is to employ a mixture of simulation and real test-bed experiments. With NS-2, we are currently implementing an NS-2 extension module that will dynamically alter EDCA parameters between VoIP sessions based on precise M2E delay information and the R-factor values. Along with our simulations, we are simultaneously building a real world wireless test-bed on which to carry out experimental tests to back up our simulation results.

4 SIMULATION

4.1 Setup

The topology for this simulation consists of running 13 simultaneous VoIP calls over an 802.11b network (Fig. 5), with an 802.11e MAC layer [5]. This number was chosen based on earlier simulations where we concluded that adding one more VoIP session to an 802.11b network would dramatically increase contention delay for all sessions.

The first test was carried out with all sessions existing within the same access category. Mean delay values are calculated for each individual session, as will happen in our final NS2 extension, and they are then plugged into a generic E-Model calculator [6] to produce a QoS R-value. This is a static calculation for the purpose of this paper, however in our implementation this will be calculated dynamically, in order to tune EDCA parameters in real time.

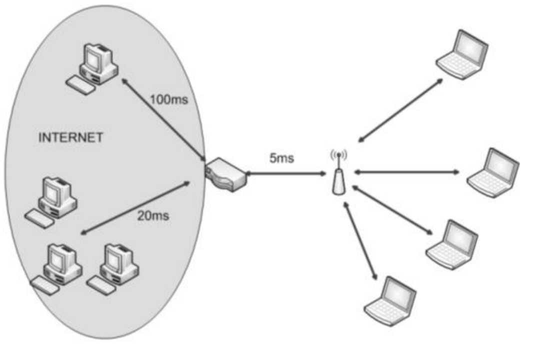


Fig. 5: NS-2 simulation topology

The first session in the simulation has a preset delay of 100ms as illustrated in Fig. 5, this is to simulate a longer geographical delay than the remaining sessions and thus a session with a lower QoS that might be improved by tuning EDCA parameters. The remaining sessions have a preset delay of 20ms on the non-wireless MAC side, and all sessions have a preset 5ms delay implemented between the AP and network router as per Fig. 5.

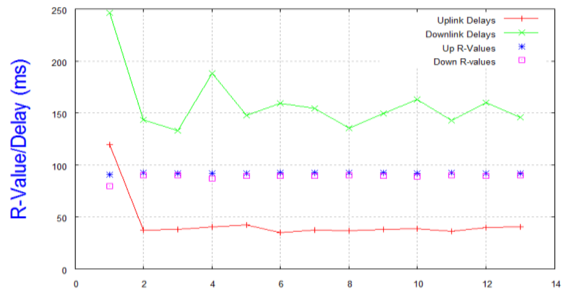
In the second test, the first session is prioritized by tuning the AIFSN of remaining traffic up by 1 to “3”, and the prioritized session maintained an AIFSN value of “2”, and all sessions had default CWmin/max values of “7” and “15” respectively.

The problem of finding the correct parameters for optimal tuning depends greatly on many factors and scenarios, this is a large research area in its own right and there is much literature published in this area, including [2]. Obtaining a lower AIFSN value than competing STAs provides some priority to a node in an 802.11e BSS, and therefore we chose this action. A more detailed investigation into existing literature may be required in future, but for the purpose of this paper we deem this action to be sufficient.

4.2 Simulation Results

In all tests, there is a major difference between uplink and downlink delays, this can be explained by a bottleneck at the AP on the downlink, where there are n stations competing for uplink wireless access against 1 AP that is channeling n times the traffic on the downlink. This problem has been referred to as “AP Throttling”, and one solution is provided in [7].

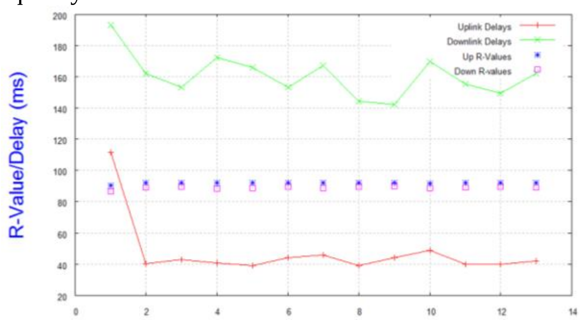
On running the first simulation (fig.6a), session 1 has a visibly higher mean delay than the remaining sessions in the network. This of course is due to the preset non-wireless MAC delay of 100ms. The mean delay due to contention was around 100ms on the downlink and 40ms on the uplink.



[session # on X axis]

Fig.6a: Mean Delay – Without Prioritization. (Associated R-Values plotted)

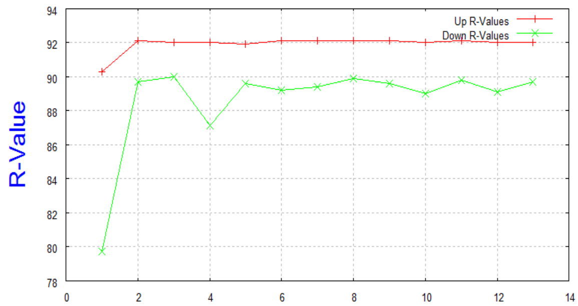
On tuning the AIFSN value for the remaining sessions and thus assigning priority to session 1, there was a noticeable improvement in the delay for session 1 (Fig. 6b). This did not come at the expense of the remaining sessions to any large degree, as the downlink traffic experienced no visible change, and the mean delays for the traffic on the uplink did increase to an extent, but not enough to have any noticeable effect on the quality of the sessions.



[session # on X axis]

Fig. 6b: Mean Delay – With Prioritization. (Associated R-Values plotted)

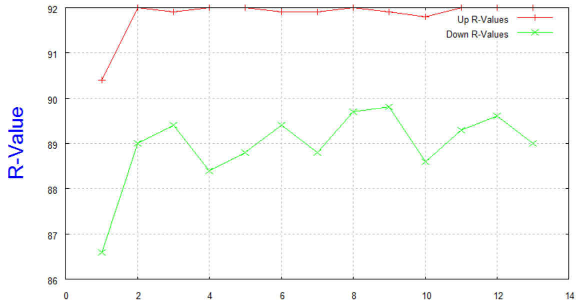
When taking a closer look the R-Values for each session, the QoS for sessions before prioritizing session 1 was *“medium, with some users dissatisfied”*, according to the E-Model, this was due to the R-Value of “79” for traffic on the downlink (Fig. 6c).



[session # on X axis]

Fig. 6c: R-Values for uplink and downlink – Without prioritization

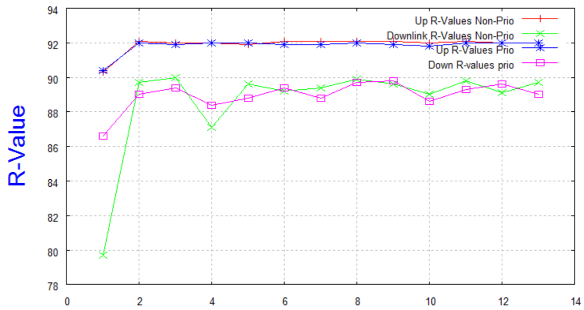
This QoS was improved dramatically where the R-Value increased to 86.5, again without much effect on remaining stations (Fig. 6d).



[session # on X axis]

Fig. 6d: R-Values for uplink and downlink – With Prioritization

Fig 7 compares the up and downlink R-Values for each session



[session # on X axis]

Fig. 7: Comparison of R-Values

The evidence of an effect on QoS of VoIP in these simulations is shown in Fig. 7. As mentioned earlier, the improvement is due to the prioritized traffic on the down link that is queued at the AP, which has a better chance of accessing the wireless medium, both against contending downlink traffic within the AP queue, and against the remaining wireless nodes.

5 CONCLUSION

In this paper we have shown how tuning EDCA parameters in an 802.11e infrastructure network can improve the Quality of Service for VoIP where a certain VoIP session has a large M2E delay and therefore degraded QoS. By optimizing EDCA parameters, the overall QoS can be improved for the given session, without imposing degradation on remaining sessions.
Although these simulations were carried out on a specific case scenario, we believe that, in addition to much existing literature showing the benefits on tuning different EDCA parameters in many different scenarios, we have shown how QoS improvements can be brought about, and on completion of our NS-2 extension, we expect to produce a comprehensive set of results validating this point.

Future work consists of continuing development of, and completing our NS-2 extension, that will dynamically tune EDCA parameters to try and implement and maintain an equalization of QoS for multiple VoIP sessions based on an E-Model graph, as outlined in section 3. We are also in the development stage of assembling a test bed where we aim to reproduce our simulation results in a real world environment.

Acknowledgments

This research is funded by Údarás na Gaeltachta, and carried out at NUI Galway city campus and at Acadamh na hOllscoilaiochta Gaeilge, Carna[8]. This research is also carried out within the Performance Engineering Laboratory (PEL)[9], a research group located at a number of universities including NUI Galway, UCD and DCU.

References

[1] ITU-T Recommendation G.107 – "*The E-Model: A Computational Model for Use in Transmission Planning*"

[2] *Experimental tuning of the AIFSN parameter to prioritize Voice over Data Transmission in 802.11e WLAN Networks*, M. Narbutt, M. Davis, IEEE International Conference on Signal Processing and Communications, Dubai, UAE, Nov. 2007

[3] The Network Simulator ns-2, Documentation, http://www.isi.edu/nsnam/ns/ns-documentation.html

[4] ITU-T Recommendation G.114 SERIES G: *Transmission Systems and Media, Digital Systems and Networks*, One way transmission time

[5] http://www.tkn.tu-berlin.de/research/802.11e_ns2/

[6] ITU-T E-Model calculation tool, http://www.itu.int/ITU-T/studygroups/com12/emodelv1/calcul.php

[7] *On improving Voice Capacity in 802.11e Infrastructure Networks*, P. Clifford, K. Duffy, D.J. Leith, D. Malone, International Conference on Wireless Networks, Communication and Mobile Computing, 2005

[8] http://www.nuigalway.ie/acadamh/

[9] http://pel.it.nuigalway.ie/